Processing 20 Million Rows Using Spark in Microsoft Fabric

Data Storage Strategy

- **Parquet**: Used for storing the large volume of log data due to its space efficiency and support for schema evolution.
- **CSV**: Selected for processed output because it is more analytics-friendly and widely supported by tools.
- **S3 Bucket**: Used as the storage location for log files, ensuring scalable and cost-efficient storage.

Pre-Requisites

1. **Set Up S3 Bucket**:
    - Create an S3 bucket named `task-log-storage`.
    - Inside the bucket, create a folder named `task_logs_2024`.
2. **Configure AWS CLI**:
    - Download and install the AWS CLI.
    - Configure user access with `getObject` and `putObject` permissions: Json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::47********55:user/<your_name>Admin"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::task-log-storage/*"
    }
  ]
}
```

3. Run the `synthetic_data_generation.py` script. The result is written to the `task_logs` folder as shown below:

Name	Date modified	Type	Size
task_log_2024-12-01_file_1.parquet	12/19/2024 12:12 AM	PARQUET File	100,792 KB
task_log_2024-12-02_file_2.parquet	12/19/2024 12:15 AM	PARQUET File	100,775 KB
task_log_2024-12-03_file_3.parquet	12/19/2024 12:18 AM	PARQUET File	100,812 KB
task_log_2024-12-04_file_4.parquet	12/19/2024 12:20 AM	PARQUET File	100,765 KB
task_log_2024-12-05_file_5.parquet	12/19/2024 12:22 AM	PARQUET File	100,766 KB
task_log_2024-12-06_file_6.parquet	12/19/2024 12:25 AM	PARQUET File	100,793 KB
task_log_2024-12-07_file_7.parquet	12/19/2024 12:27 AM	PARQUET File	100,752 KB
task_log_2024-12-08_file_8.parquet	12/19/2024 12:29 AM	PARQUET File	100,784 KB
task_log_2024-12-09_file_9.parquet	12/19/2024 12:31 AM	PARQUET File	100,778 KB
task_log_2024-12-10_file_10.parquet	12/19/2024 12:33 AM	PARQUET File	100,797 KB
task_log_2024-12-11_file_11.parquet	12/19/2024 12:35 AM	PARQUET File	100,774 KB
task_log_2024-12-12_file_12.parquet	12/19/2024 12:38 AM	PARQUET File	100,774 KB
task_log_2024-12-13_file_13.parquet	12/19/2024 12:40 AM	PARQUET File	100,795 KB
task_log_2024-12-14_file_14.parquet	12/19/2024 12:42 AM	PARQUET File	100,788 KB
task_log_2024-12-15_file_15.parquet	12/19/2024 12:44 AM	PARQUET File	100,804 KB

4. Test permissions by uploading log files:

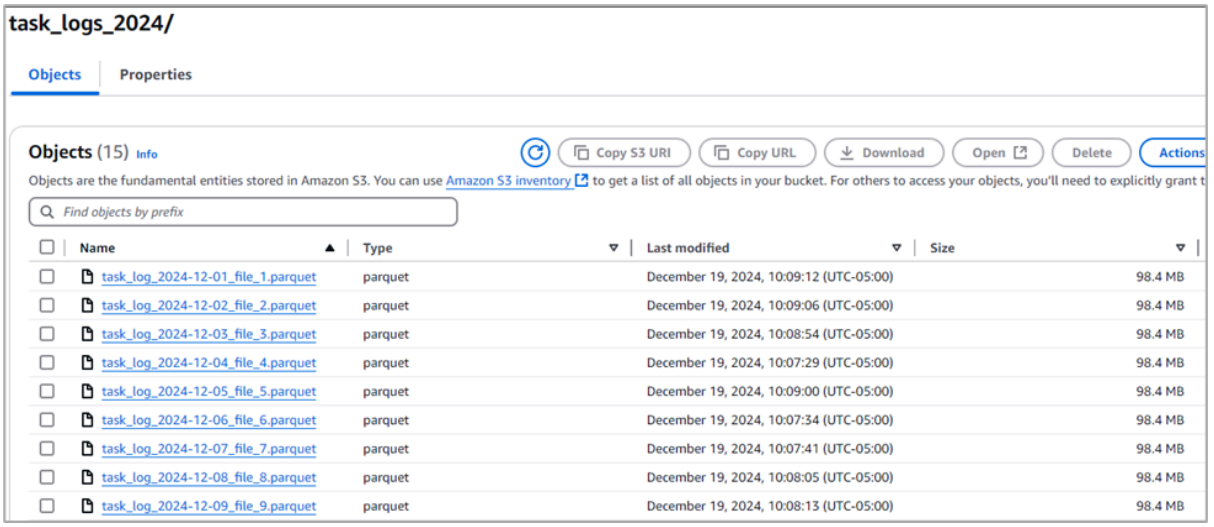
   Run: aws s3 cp task_logs/ s3://task-log-storage/task_logs_2024/ --recursive
   Result:

```
C:\Users\olanr\Desktop\data_science\sora_union\Question_3>dir
 Volume in drive C is Windows
 Volume Serial Number is C6C0-1322

 Directory of C:\Users\olanr\Desktop\data_science\sora_union\Question_3

12/19/2024  12:09 AM    <DIR>          .
12/18/2024  12:08 PM    <DIR>          ..
12/18/2024  09:56 PM                92 aws_credentials.env
12/18/2024  11:30 PM    <DIR>          images
12/19/2024  12:09 AM             2,053 synthetic_data_generation.py
12/18/2024  08:03 PM    <DIR>          task_logs
               2 File(s)          2,145 bytes
               4 Dir(s)  39,816,785,920 bytes free

C:\Users\olanr\Desktop\data_science\sora_union\Question_3>aws s3 cp task_logs/ s3://task-log-storage/task_logs_2024/ --r
ecursive
upload: task_logs\task_log_2024-12-04_file_4.parquet to s3://task-log-storage/task_logs_2024/task_log_2024-12-04_file_4.
parquet
upload: task_logs\task_log_2024-12-06_file_6.parquet to s3://task-log-storage/task_logs_2024/task_log_2024-12-06_file_6.
parquet
upload: task_logs\task_log_2024-12-07_file_7.parquet to s3://task-log-storage/task_logs_2024/task_log_2024-12-07_file_7.
```


task_logs_2024/
Objects
Properties
Objects (15) Info
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant t
Find objects by prefix
Name
Type
Last modified
Size
task_log_2024-12-01_file_1.parquet parquet December 19, 2024, 10:09:12 (UTC-05:00) 98.4 MB
task_log_2024-12-02_file_2.parquet parquet December 19, 2024, 10:09:06 (UTC-05:00) 98.4 MB
task_log_2024-12-03_file_3.parquet parquet December 19, 2024, 10:08:54 (UTC-05:00) 98.4 MB
task_log_2024-12-04_file_4.parquet parquet December 19, 2024, 10:07:29 (UTC-05:00) 98.4 MB
task_log_2024-12-05_file_5.parquet parquet December 19, 2024, 10:09:00 (UTC-05:00) 98.4 MB
task_log_2024-12-06_file_6.parquet parquet December 19, 2024, 10:07:34 (UTC-05:00) 98.4 MB
task_log_2024-12-07_file_7.parquet parquet December 19, 2024, 10:07:41 (UTC-05:00) 98.4 MB
task_log_2024-12-08_file_8.parquet parquet December 19, 2024, 10:08:05 (UTC-05:00) 98.4 MB
task_log_2024-12-09_file_9.parquet parquet December 19, 2024, 10:08:13 (UTC-05:00) 98.4 MB

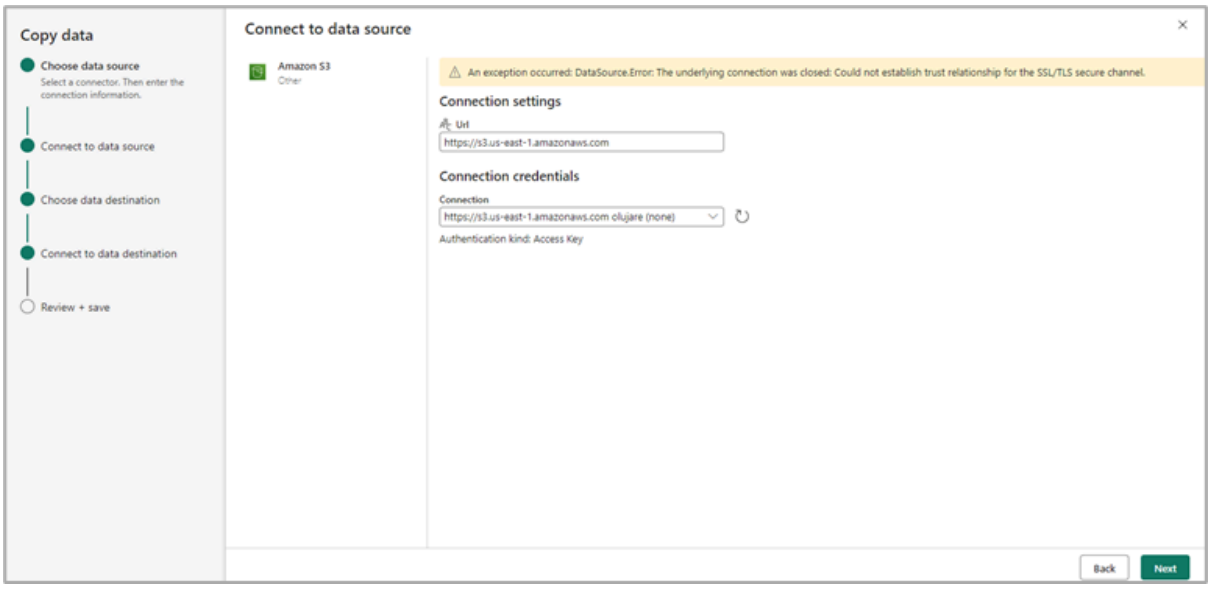
Data Processing Workflow

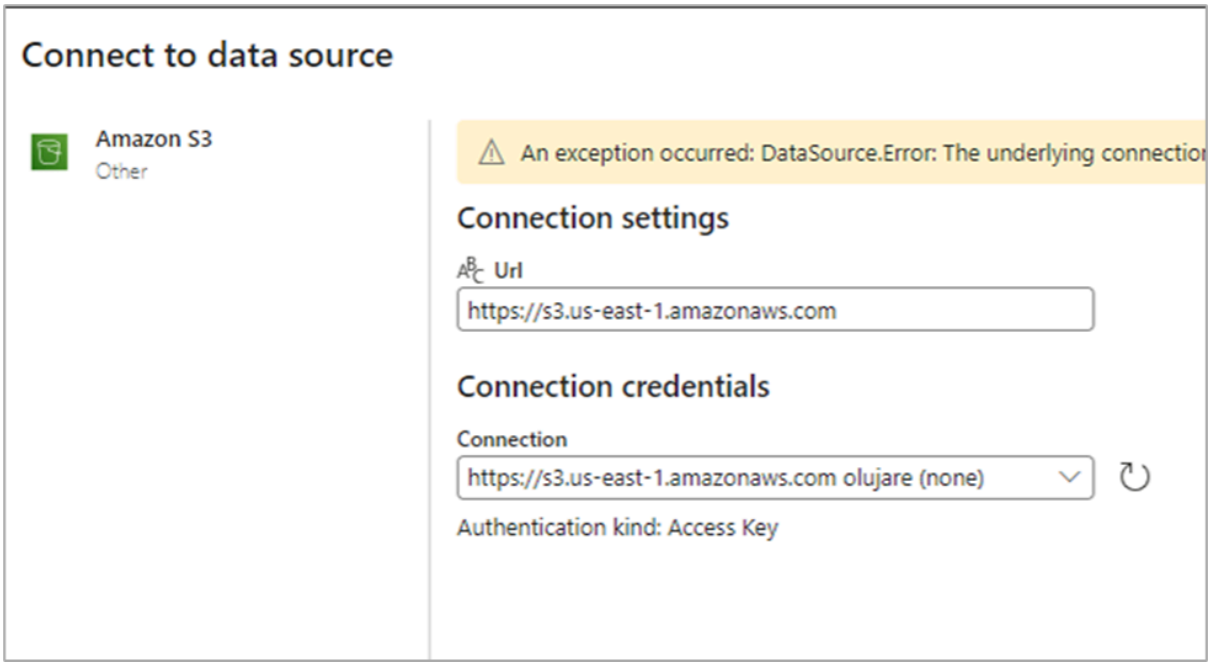
1. Set Up Microsoft Fabric

- Navigate to **Microsoft Fabric** and create a **new workspace**.
- Within the workspace, create a **new data pipeline**. This pipeline will be used to load data from the S3 bucket to a Fabric Lakehouse.

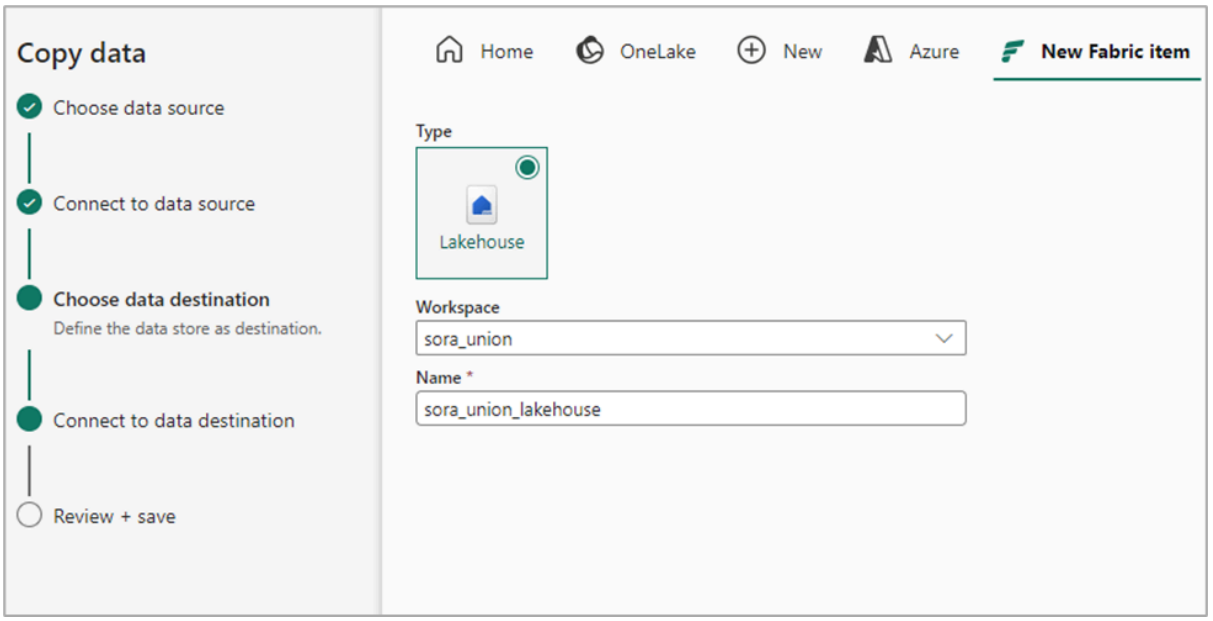
2. Create a Data Pipeline

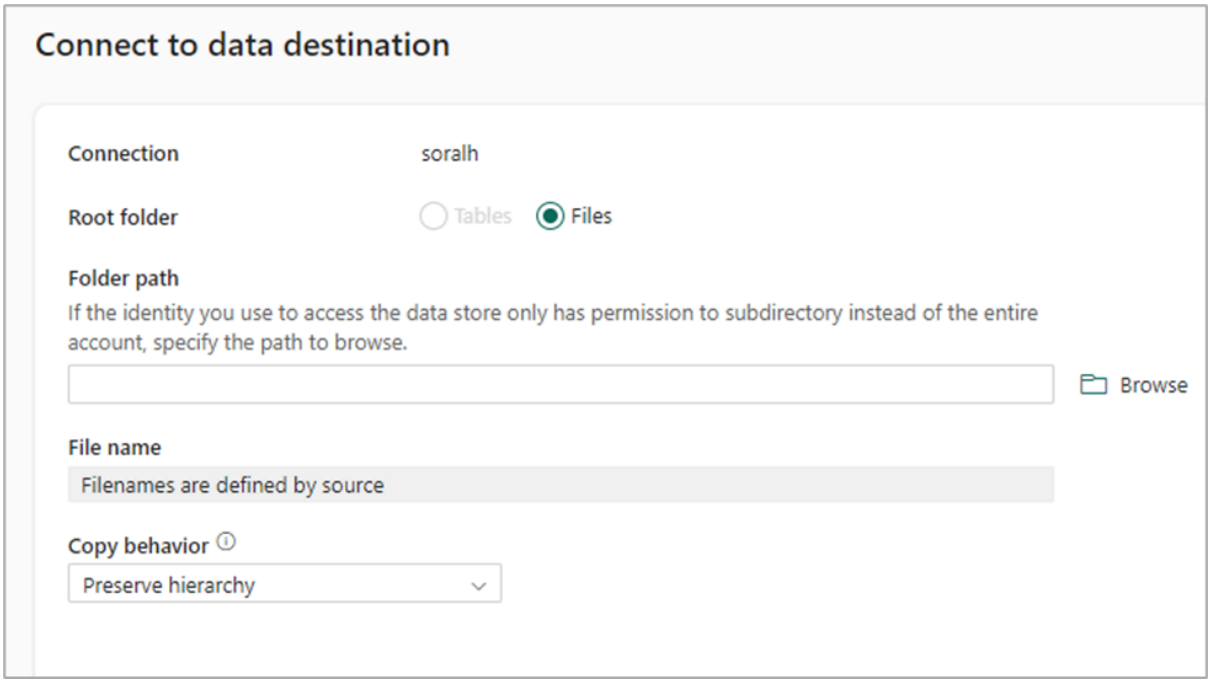
- Follow the instructions in the official [Microsoft Fabric Guide]() to create and configure the pipeline.
- **Connect to the Data Source**:



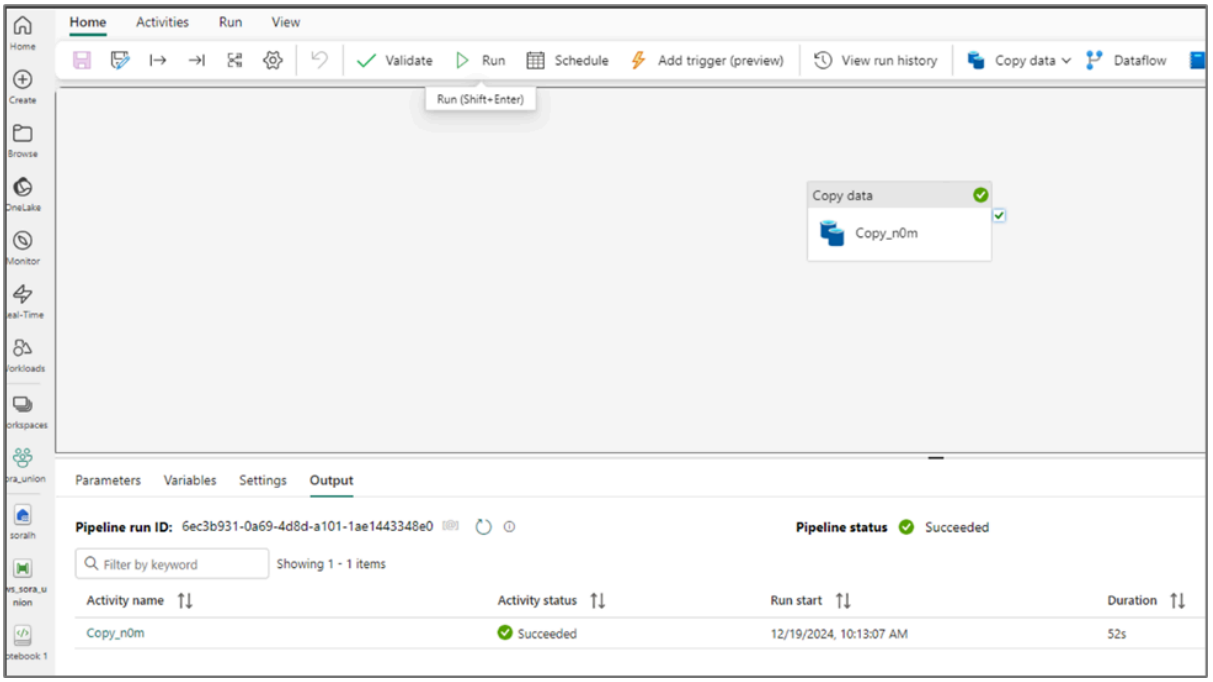


    - Select the S3 bucket as the source.
    - Enable schema-agnostic mode (binary copy) since the input format is Parquet.
- **Set Up the Destination**:
    - Under the **New Fabric Item** tab, create a **new lakehouse** and configure it as the destination.





- Run the pipeline to load the Parquet data into the Fabric Lakehouse.



3. Set Up a Spark Notebook

- Create a new **Spark Notebook** in Fabric.
- Grant the notebook access to the Fabric Lakehouse for seamless data access.

4. Process Data Using Spark

- Write Spark code to:
  - Load the Parquet data from the Lakehouse.
  - Perform transformations, aggregations, or filtering as required.
  - Save the processed data as a CSV file back to the Lakehouse or another destination.